

# PreSTU: Pre-Training for Scene-Text Understanding (Supplementary Material)

Jihyung Kil<sup>1\*</sup> Soravit Changpinyo<sup>2</sup>  
Xi Chen<sup>2</sup> Hexiang Hu<sup>2</sup> Sebastian Goodman<sup>2</sup> Wei-Lun Chao<sup>1</sup> Radu Soricut<sup>2</sup>  
<sup>1</sup>The Ohio State University <sup>2</sup>Google Research  
{kil.5, chao.209}@osu.edu  
{schangpi, chillxichen, hexiang, seabass, rsoricut}@google.com

## Appendices

In this supplementary material, we provide details omitted in the main text.

- **Appendix A:** V&L model implementation details (cf. §2.1 of the main text).
- **Appendix B:** Pre-training & Scene-text V&L datasets (cf. §2.2.2 & §2.3 of the main text).
- **Appendix C:** More comparisons to prior works (cf. §3.1.4 of the main text).
- **Appendix D:** More ablation studies (cf. §3.2 of the main text).
- **Appendix E:** Qualitative results.
- **Appendix F:** Contributions.

## A. V&L model implementation details

Our model is an encoder-decoder V&L architecture consisting of ViT-B/16 [7] as a visual module and mT5-Base [33] as a language module. For the vision module, we adopt a transformer-based vision model ViT [7] pre-trained on JFT-3B dataset [35], the extension of JFT-300M [28], with 3 billion images collected from the web. Our language module is initialized from mT5-Base [33], a multilingual variant of T5 [23], pre-trained on a new Common Crawl-based dataset with 101 different languages.

During training, all parameters in vision and language blocks are updated simultaneously. We choose Adafactor [25] as an optimizer with  $\beta_1 = 0$  and second-moment exponential decay = 0.8. For a learning rate, we schedule a linear warmup for 1K steps with inverse square-root decay. Our V&L architecture is implemented in Jax/Flax [4] based on the open-source T5X [24] framework.

We have done extensive hyperparameter tuning for our experiments. For instance, we find that the best hyper-

parameter configuration for SPLITOCR pre-training is — initial (peak) learning rate: 1e-3, batch size: 256, image resolution: 640x640, the length of input/target text tokens: 40/26, and dropout: 0.1. For TextVQA, we achieve the best result with initial learning rate: 2e-4 and the length of input/target text tokens: 72/8 (See Table A for more details).

## B. Pre-training & Scene-text V&L datasets

We provide more details about pre-training and scene-text V&L datasets used in our experiments.

**Scene-Text on CC15M.** We estimate the portion of scene text on CC15M with a study on 300 randomly sampled images. We manually check each image and found: 59% (177/300) have scene text; only 13% (38/300) are watermark-only images. This aligns with TAP’s report [34] on CC3M (scene-text: 42%, watermark-only: 5%). Note that TAP mentioned “*only the CC dataset contains a reasonable portion of images with meaningful scene text regions*”, suggesting CC15M is suitable for STU pre-training.

**ST-VQA** [3] is for scene-text VQA dataset. Its images are collected from various resources: COCO-Text [29], Visual Genome [17], VizWiz [9], ICDAR [14, 13], ImageNet [6], and IIIT-STR [21]. Since there is no official validation set, we follow the split provided by M4C [11], resulting in 23K/26K training/validation VQA examples.

**TextVQA** [27] for scene-text VQA. It is a subset of Open Images [16] with scene-text related QA pairs from human annotators with ten ground-truth answers. It has 34K/5K training/validation VQA examples from 21K/3K images.

**VizWiz-VQA** [9]. The dataset contains 20K/3K training/validation VQA examples collected from blind users. Due to the nature of the questions asked by blind people, we identify this benchmark as a candidate to benefit from scene-text understanding, even though it was not directly designed for scene-text VQA.

**VQAv2** [8]. We further evaluate PRESTU on standard VQA benchmark to check if the scene-text recognition can

\* Work done at Google Research.

Hyper-parameter	Pre-training		Downstream				
	SPLITOCR	ST-VQA	TextVQA	VizWiz-VQA	VQAv2	TextCaps	VizWiz-Caption
Initial (peak) learning rate	1e-3	9e-4	2e-4	9e-4	1e-3	2e-4	2e-4
Batch size	256	256	256	256	512	256	256
Image resolution	640x640	640x640	640x640	640x640	640x640	640x640	640x640
Length of input text tokens	40	72	72	72	72	56	56
Length of target text tokens	26	8	8	8	8	64	64
Dropout	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table A: **Best hyper-parameters for our experiments.** Among hyper-parameters of our V&L model, we find that initial (peak) learning rate, batch size, image resolution, length of input/target text tokens, and dropout are major components affecting the performance of our tasks.

Model	Vision / Language	Model Size	Data Size	Pre-training Objective	Scene-text V&L Benchmark					
					ST-VQA ANLS	TextVQA Acc	VizWiz-VQA Acc	VQAv2 Acc	TextCaps CIDEr	VizWiz-Captions CIDEr
NOPRESTU	ViT-B16 / mT5 <sub>base</sub>	473M	0	-	56.7 (55.6)	44.8 (45.2)	57.2 (57.7)	75.2 (74.8)	96.9 (100.0)	87.2 (87.7)
PRESTU	ViT-B16 / mT5 <sub>base</sub>	473M	13M	VQA/CAP	N/A (N/A)	48.3 (47.2)	57.6 (58.3)	75.0 (75.0)	133.1 (130.2)	103.1 (103.6)
				SPLITOCR	65.5 (62.7)	55.2 (55.6)	61.3 (61.9)	76.2 (76.0)	126.1 (134.6)	90.2 (90.3)
				SPLITOCR→VQA/CAP	N/A (N/A)	56.3 (56.7)	62.0 (62.5)	76.1 (76.1)	139.1 (141.7)	105.6 (105.6)
TAP [34]	FRCNN / BERT <sub>base</sub>	146M	1.5M	MLM+ITM+RPP	59.7 (59.8)	54.0 (54.7)	- (-)	- (-)	109.7 (119.0)	- (-)
LaTr [2]	ViT-B/16 / T5 <sub>large</sub>	831M	64M	MLM	69.6 (70.2)	61.6 (61.1)	- (-)	- (-)	- (-)	- (-)
Flamingo [1]	NFNet / Chinchilla	80B	2.3B	VLM	- (-)	54.1 (57.1)	65.4 (65.7)	82.1 (82.0)	- (-)	- (-)
GIT <sub>L</sub> [31]	CoSwin / TransDec	347M	20M	VLM	- (44.6)	- (37.5)	- (62.5)	- (75.5)	- (106.3)	- (96.1)
GIT2 [31]	DaViT / TransDec	5B	12.9B	VLM	75.8 (75.1)	67.3 (68.4)	70.1 (71.0)	81.9 (81.7)	145.0 (148.6)	120.8 (119.4)
PaLI [5]†	ViT-e / mT5-XXL	16B	10B	our OCR w/ others	79.9 (-)	73.1 (71.8)	73.3 (74.4)	84.3 (84.3)	160.4 (160.0)	- (-)

Table B: **Full Comparison to prior works.** FRCNN: Faster R-CNN, TransDec: 6-layer transformer decoder, MLM: Masked Language (visual region) Modeling, ITM: Image-Text Matching, RPP: Relative Position Prediction, VLM: Visual Language Modeling. Following [31], the parameters of text token embeddings are not counted in the model size. We report results on the **test (validation)** set for ST-VQA, the **test-std (validation)** for TextVQA/TextCaps, and the **test-std (test-dev)** set for VizWiz-VQA, VQAv2, and VizWiz-Captions. †: our objective OCR is an ingredient in their pre-training objectives.

also help on general VQA tasks. Following [12], we use the VQAv2 train/dev splits of \*train2014/minival2014, which are 592K/65K VQA examples in total.

**TextCaps** [26] for scene-text image captioning task. It uses the same subset of OpenImages images with TextVQA. Each image has five ground-truth captions, totaling 100K/15K training/validation captions.

**VizWiz-Captions** [10]. Like Vizwiz-VQA, this benchmark was generated by blind users to solve their daily visual challenges. It contains 23.4K/7.7K training/validation images, where each image is paired with five captions. In total, there are 117K/38K training/validation image captions.

**OCR-VQA** [22] is an OCR-based VQA dataset about images of book covers. Concretely, it requires models to answer visual questions by reading/interpreting the text on the book covers (e.g., author, title). In summary, OCR-VQA provides 207K images of book covers and more than 1 million VQA examples.

**DocVQA** [20] asks for the textual (handwritten, type-written, printed) content on the document images. In contrast with general VQA [8], models should understand addi-

tional visual cues, including layout (e.g., tables), style (e.g., font, color), and non-textual elements (e.g., tick boxes). In total, DocVQA contains 50K VQA examples with more than 12K document images.

**ChartQA** [19] is a VQA benchmark based on charts. Specifically, it covers more than 23K VQA examples from 17K charts. In ChartQA, models are required to perform complex reasoning (e.g., logical and arithmetic operations) to understand charts and the corresponding questions.

**AI2D** [15] is a VQA dataset of illustrative diagrams. The task of AI2D is to answer diagram-related questions by analyzing the diagram structure and identifying its visual entities and their semantic relationships. AI2D provides 5K diagrams with 15K VQA examples in total.

**WidgetCap** [18] aims to generate language descriptions for UI elements (widgets) in the mobile interface. Mobile apps often lack widget captions in their interfaces, which recently becomes a primary issue for mobile accessibility. WidgetCap attempts to solve this challenge by providing an evaluation benchmark containing more than 162K language phrases (i.e., captions) with 61K UI elements.

**Screen2Words** [30] is an image captioning task to generate a short summary of the mobile screen. To complete the task, models should have the capability of understanding the screen and conveying its content and functionalities in a concise language phrase. Screen2Words consists of 112K captions for 22K mobile screens in total.

### C. More comparisons to prior works

**Comparison to TAP.** While PRESTU adopts a *general* pre-training dataset (*i.e.*, CC15M), TAP’s pre-training data aggregates scene-text *dedicated* downstream data, including ST-VQA, TextVQA, TextCaps, and OCR-CC. Thus, even if the size of TAP’s pre-training data (1.5M) is smaller, it may align better with the downstream tasks. However, since TAP’s approach focuses on the specific downstream tasks, it is less applicable to other V&L tasks, whereas PRESTU provides a more flexible interface.

Moreover, TAP adopts closed-set prediction by training an answer classifier based on the dataset-specific vocabulary. This may benefit the accuracy of the corresponding downstream task. In contrast, PRESTU chooses open-ended prediction as it is more generalizable in practice and is adopted by many recent works (*e.g.*, PaLI, GIT).

**Full Comparison.** Table B shows full comparisons to prior works on all splits of benchmarks. Concretely, we report results on the **test (validation)** set for ST-VQA, the **test-std (validation)** for TextVQA/TextCaps, and the **test-std (test-dev)** set for VizWiz-VQA, VQAv2, and VizWiz-Captions. Aligned with the results in the main text, SPLITOCR outperforms NOPRESTU on all evaluation metrics. In addition, SPLITOCR→VQA/CAP further boosts the performance, highlighting the importance of task-specific objectives (VQA and CAP) during pre-training.

### D. More ablation studies

**SPLITOCR vs. CAP.** Table 1 of the main text shows the effectiveness of SPLITOCR against VQA on VQA tasks. We further check its benefit over CAP on VQA tasks. As shown in Table C, SPLITOCR consistently improves over CAP (*e.g.*, 53.2% vs. 49.3%) on TextVQA, further supporting that SPLITOCR is important for higher accuracy.

We also investigate the effect of the order of pre-training stages. Concretely, we switch the order between SPLITOCR and CAP and demonstrate that applying SPLITOCR first (*i.e.*, default setting) is better (Table D).

**Order of OCR.** PRESTU uses the fixed OCR order to standardize the target output sequence during pre-training. Compared to the random order, we see its advantage with consistent improvements (*e.g.*, 132.4 vs. 134.6 on TextCaps CIDEr / 55.3% vs. 55.6% on TextVQA).

**OCR System.** We note that different prior works often use different *commercial* OCR engines to obtain their best






Image	gOCR token	PRESTU OCR token prediction
	panera bread drive thru	panera bread drive thru
	north course par 4 353 333 287 hdcp - 13-15	north course par 4 333 333 287
	a 4005 ealing	a 4005 ealing
	sk - ii facial treatment essence	sk - ii facial treatment essence

Figure A: **PRESTU’s OCR token prediction.** The quality of OCR tokens generated by SPLITOCR is comparable to that of gOCR system. This shows the possibility of leveraging SPLITOCR as an alternative OCR system when other systems are not available.



**TextVQA**

what player number is the runner sliding under?


**Ground-truth:** 13

**gOCR tokens:** machaden

**NoPreSTU (Baseline):** 5

**PRESTU:** 13

---



**TextVQA**

what is the make of car?

**Ground-truth:** lexus

**gOCR tokens:** cooper stu  
lexue ecnk-06n

**NoPreSTU (Baseline):** cooper

**PRESTU:** lexus

Figure B: **gOCR tokens vs. PRESTU prediction on TextVQA.** gOCR system does not detect some OCR tokens in the image (*e.g.*, “13”) or detects them incorrectly (*e.g.*, “lexue”). This leads NOPRESTU to predict wrong answers (*e.g.*, “5” or “cooper”). On the other hand, SPLITOCR with gOCR tokens as input predicts the answers correctly with correct OCR tokens (*e.g.*, “13” or “lexus”).

results. Thus, it is hard to perform a fair comparison without extra costs. That said, we did evaluate PRESTU with different OCR engines (including Rosetta-en) at the downstream stage (Table 10 of the main text). A similar setup is used in LaTr [2]: Rosetta-en/Amazon-OCR for downstream TextVQA/pre-training, respectively. In this setup, PRESTU outperforms LaTr on TextVQA Val (50.7% vs. 48.4%).

Model	Pre-training Objective	TextVQA Val Acc
PRESTU	CAP	49.3
	SPLITOCR→CAP	53.2
	CAP→VQA	50.0
	SPLITOCR→CAP→VQA	55.0

Table C: SPLITOCR vs. CAP on VQA tasks. SPLITOCR is crucial for higher accuracy.

Model	Pre-training Objective	TextCaps Val CIDEr
PRESTU	SPLITOCR→CAP	141.7
	CAP→SPLITOCR	135.4

Table D: Effect of switching pre-training stages. Applying SPLITOCR first (*i.e.*, default setting) is more effective.

## E. Qualitative results

Figure A shows some examples of OCR tokens generated by SPLITOCR. Our SPLITOCR detects all (or almost all) OCR tokens in the images correctly, competitive to the gOCR system.

In §3.2 of the main text, we demonstrate that having two sources of OCR signals is beneficial (OCR signals by pre-trained ViT with SPLITOCR and OCR signals by gOCR system). Figure B further supports this finding qualitatively. For instance, gOCR alone does not detect some OCR tokens in the image (*e.g.*, “13”) or detects them incorrectly (*e.g.*, “lexue”). This leads NOPRESTU to predict wrong answers (*e.g.*, “5” or “cooper”). On the other hand, SPLITOCR with gOCR tokens as input predicts the answers correctly with correct OCR tokens (*e.g.*, “13” or “lexus”), demonstrating that two sources of OCR signals (*i.e.*, ViT and gOCR) are complementary.

Figure C provides qualitative results for VizWiz-VQA and VizWiz-Captions, demonstrating the applicability of PRESTU to different VQA and image captioning tasks.

## F. Contributions

While our SPLITOCR is inspired by SimVLM [32], the motivation is fundamentally different and it is not trivial to apply the prefix idea in the first place for OCR-aware pre-training. Concretely, SimVLM aims to serve downstream tasks that generate text like captions or answers (with optional text input). Thus, it is understandable why SimVLM could help. In contrast, for downstream STU tasks, *OCR strings often serve only as the text input (Figures 2 & 3 of the main text)*. Therefore, while it makes sense to apply our second stage pre-training (CAP & VQA) with OCR strings as the input, it is not intuitive to develop a separate OCR-only pre-training stage (SPLITOCR) that leverages the idea

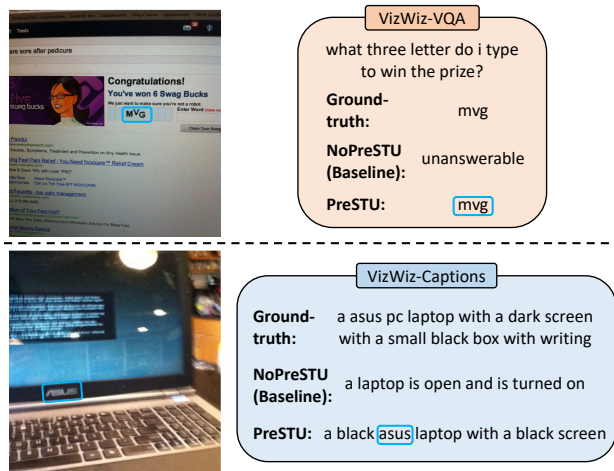


Figure C: Qualitative results on VizWiz-VQA [9] and VizWiz-Captions [10].

of SimVLM. We came up with SPLITOCR purely from the two essential STU capabilities: (i) recognizing text in an image, (ii) connecting the text to its visual context. Our contribution thus lies in how to fulfill the two requirements via a unified manner, which turns out to be a SimVLM-like objective.

Besides SPLITOCR, another key contribution of our work is the comprehensive investigation of pre-training STU capabilities using a combination of easily reproducible objectives and a standard network architecture, on domains much more diverse than in previous works. Thus, we believe that our extensive analysis is valuable to the community.

Finally, we demonstrate the effectiveness of our OCR-aware method in large-scale settings. We choose CC15M as pre-training dataset, which is often considered large-scale, and PaLI [5], an extremely large-scale model (with 10B data), utilizes our objective to achieve SOTA results on nearly all STU tasks (cf. §3.1.4 of the main text). This shows the utility of our pre-training objectives even in SOTA large-scale models.

## References

- [1] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikolaj Binkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: a visual language model for few-shot learning. In *NeurIPS*, 2022. 2
- [2] Ali Furkan Biten, Ron Litman, Yusheng Xie, Srikar Appalaraju, and R Manmatha. Latr: Layout-aware transformer for scene-text vqa. In *CVPR*, 2022. 2, 3



- [3] Ali Furkan Biten, Ruben Tito, Andres Mafla, Lluís Gomez, Marçal Rusinol, Ernest Valveny, C.V. Jawahar, and Dimosthenis Karatzas. Scene text visual question answering. In *ICCV*, 2019. 1
- [4] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, et al. Jax: composable transformations of python+ numpy programs. *Version 0.2*, 2018. 1
- [5] Xi Chen, Xiao Wang, Soravit Changpinyo, AJ Piergiovanni, Piotr Padlewski, Daniel Salz, Sebastian Alexander Goodman, Adam Grycner, Basil Mustafa, Lucas Beyer, Alexander Kolesnikov, Joan Puigcerver, Nan Ding, Keran Rong, Hassan Akbari, Gaurav Mishra, Linting Xue, Ashish Thapliyal, James Bradbury, Weicheng Kuo, Mojtaba Seyedhosseini, Chao Jia, Burcu Karagol Ayan, Carlos Riquelme, Andreas Steiner, Anelia Angelova, Xiaohua Zhai, Neil Houlsby, and Radu Soricut. PaLI: A jointly-scaled multilingual language-image model. In *ICLR*, 2023. 2, 4
- [6] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, 2009. 1
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021. 1
- [8] Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the v in vqa matter: Elevating the role of image understanding in visual question answering. In *CVPR*, 2017. 1, 2
- [9] Danna Gurari, Qing Li, Abigale J. Stangl, Anhong Guo, Chi Lin, Kristen Grauman, Jiebo Luo, and Jeffrey P. Bigham. VizWiz Grand Challenge: Answering visual questions from blind people. In *CVPR*, 2018. 1, 4
- [10] Danna Gurari, Yinan Zhao, Meng Zhang, and Nilavra Bhattacharya. Captioning images taken by people who are blind. In *ECCV*, 2020. 2, 4
- [11] Ronghang Hu, Amanpreet Singh, Trevor Darrell, and Marcus Rohrbach. Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In *CVPR*, 2020. 1
- [12] Yu Jiang, Vivek Natarajan, Xinlei Chen, Marcus Rohrbach, Dhruv Batra, and Devi Parikh. Pythia v0.1: the winning entry to the VQA challenge 2018. *arXiv preprint arXiv:1807.09956*, 2018. 2
- [13] Dimosthenis Karatzas, Lluís Gomez-Bigorda, Angelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, 2015. 1
- [14] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluís Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluís Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, 2013. 1
- [15] Aniruddha Kembhavi, Mike Salvato, Eric Kolve, Minjoon Seo, Hannaneh Hajishirzi, and Ali Farhadi. A diagram is worth a dozen images. In *ECCV*, 2016. 2
- [16] Ivan Krasin, Tom Duerig, Neil Alldrin, Vittorio Ferrari, Sami Abu-El-Hajja, Alina Kuznetsova, Hassan Rom, Jasper Uijlings, Stefan Popov, Andreas Veit, et al. Openimages: A public dataset for large-scale multi-label and multi-class image classification. *Dataset available from <https://github.com/openimages>*, 2017. 1
- [17] Ranjay Krishna, Yuke Zhu, Oliver Groth, Justin Johnson, Kenji Hata, Joshua Kravitz, Stephanie Chen, Yannis Kalantidis, Li-Jia Li, David A Shamma, et al. Visual genome: Connecting language and vision using crowdsourced dense image annotations. In *IJCV*, 2017. 1
- [18] Yang Li, Gang Li, Luheng He, Jingjie Zheng, Hong Li, and Zhiwei Guan. Widget captioning: generating natural language description for mobile user interface elements. *arXiv preprint arXiv:2010.04295*, 2020. 2
- [19] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *arXiv preprint arXiv:2203.10244*, 2022. 2
- [20] Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. Docvqa: A dataset for vqa on document images. In *WACV*, 2021. 2
- [21] Anand Mishra, Karteek Alahari, and C.V. Jawahar. Image retrieval using textual cues. In *ICCV*, 2013. 1
- [22] Anand Mishra, Shashank Shekhar, Ajeet Kumar Singh, and Anirban Chakraborty. Ocr-vqa: Visual question answering by reading text in images. In *ICDAR*, 2019. 2
- [23] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. In *JMLR*, 2020. 1
- [24] Adam Roberts, Hyung Won Chung, Anselm Levskaya, Gaurav Mishra, James Bradbury, Daniel Andor, Sharan Narang, Brian Lester, Colin Gaffney, Afroz Mohiuddin, et al. Scaling up models and data with t5x and seqio. *arXiv preprint arXiv:2203.17189*, 2022. 1
- [25] Noam Shazeer and Mitchell Stern. Adafactor: Adaptive learning rates with sublinear memory cost. In *ICLR*, 2018. 1
- [26] Oleksii Sidorov, Ronghang Hu, Marcus Rohrbach, and Amanpreet Singh. TextCaps: a dataset for image captioning with reading comprehension. In *ECCV*, 2020. 2
- [27] Amanpreet Singh, Vivek Natarajan, Meet Shah, Yu Jiang, Xinlei Chen, Dhruv Batra, Devi Parikh, and Marcus Rohrbach. Towards VQA models that can read. In *CVPR*, 2019. 1
- [28] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *ICCV*, 2017. 1
- [29] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016. 1
- [30] Bryan Wang, Gang Li, Xin Zhou, Zhouong Chen, Tovi Grossman, and Yang Li. Screen2words: Automatic mobile

- ui summarization with multimodal learning. In *The 34th Annual ACM Symposium on User Interface Software and Technology*, 2021. [3](#)
- [31] Jianfeng Wang, Zhengyuan Yang, Xiaowei Hu, Linjie Li, Kevin Lin, Zhe Gan, Zicheng Liu, Ce Liu, and Lijuan Wang. GIT: A generative image-to-text transformer for vision and language. *arXiv preprint arXiv:2205.14100*, 2022. [2](#)
- [32] Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. SimVLM: Simple visual language model pretraining with weak supervision. In *ICLR*, 2022. [4](#)
- [33] Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. mT5: A massively multilingual pre-trained text-to-text transformer. In *NAACL*, 2021. [1](#)
- [34] Zhengyuan Yang, Yijuan Lu, Jianfeng Wang, Xi Yin, Dinei Florencio, Lijuan Wang, Cha Zhang, Lei Zhang, and Jiebo Luo. TAP: Text-aware pre-training for text-vqa and text-caption. In *CVPR*, 2021. [1](#), [2](#)
- [35] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *CVPR*, 2022. [1](#)